CLAIMS

1. A cache memory preprocessor, for preparing a cache memory for use by a processor, said processor being arranged to access a main memory via data caching in said associative memory, said cache memory preprocessor comprising:

a command inputter, for receiving a multiple-way cache memory processing command from said processor; and

a command implementer associated with said command inputter, for performing background processing upon multiple ways of said cache memory in accordance with said multiple-way cache memory processing command.

2. A cache memory preprocessor according to claim 1, wherein said multiple-way cache memory processing command is a block update command and said command implementer comprises a block updater operable to implement said block update command upon a specified block of ways of said cache memory by updating main memory data in accordance with data cached in said specified block of ways.

3. A cache memory preprocessor according to claim 2, wherein said block updater comprises:

a way checker, for determining for a given way of said cache memory if data cached in said way is equivalent to the corresponding main memory data; and

a data storer associated with said way checker, for carrying out said updating, when said data is not equivalent.

4. A cache memory preprocessor according to claim 1, wherein said cache memory comprises an n-way set associative memory.

5. A cache memory preprocessor according to claim 1, wherein said cache memory comprises a fully associative memory.

6. A cache memory preprocessor according to claim 1, wherein said cache memory comprises a direct mapped cache memory.

7. A cache memory preprocessor according to claim 3, wherein at least one way of said cache memory comprises a dirty bit to indicate said equivalence status and said way checker is operable to examine said dirty bit of said way.

8. A cache memory preprocessor according to claim 3, wherein said data storer is operable to store said updated data in said main memory at a main memory address associated with said cached data.

9. A cache memory preprocessor according to claim 7, wherein said data storer is further operable to reset said dirty bit, when updated data from said way is stored in said main memory.

10. A cache memory preprocessor according to claim 1, wherein said multiple-way cache memory processing command is a block invalidate command and said command implementer comprises a block invalidator operable to implement said block invalidate command upon a specified block of ways of said cache memory by invalidating the data in said specified block of ways.

11. A cache memory preprocessor according to claim 10, wherein at least one way comprises a validity bit to indicate a validity status of said way, and wherein said invalidating comprises setting a validity bit of said way to invalid.

12. A cache memory preprocessor according to claim 1, wherein said multiple-way cache memory processing command is a block initialize command and said command implementer comprises a cache initializer operable to implement said block initialize command upon a specified block of said main memory, by caching main memory data of said specified block of main memory into said cache memory.

13. A cache memory preprocessor according to claim 12, wherein said cache initializer comprises:

a cache checker, for determining if data from a selected main memory address is present in said cache memory; and

a data cacher, for carrying out said data caching, if said data is not found to be present.

14. A cache memory preprocessor according to claim 1, wherein said cache memory preprocessor is configured to operate with a segmented memory having a plurality of main memory segments, and wherein data caching is provided to each of said main memory segments by a cache memory section within said memory segment.

15. A cache memory preprocessor according to claim 14, connectable to said cache memory sections via an interconnector, the interconnector providing in parallel switchable connections between each of a plurality of processing agents to selectable ones of said cache memories.

16. A cache memory preprocessor according to claim 15, wherein said interconnector comprises a prioritizer operable to prevent simultaneous connection of more than one output to a memory segment by controlling access to said cache memory sections according to a priority scheme.

17. A background memory refresher, for updating main memory data in accordance with data cached in a cache memory, wherein said cache memory is arranged in blocks, comprising:

a command inputter, for receiving a block update command; and

a block updater, associated with said command inputter, for performing background update operations blockwise from a specified block of said cache memory so as to update said main memory in accordance with data cached in said specified block of said cache memory.

18. A background memory refresher according to claim 17, wherein said cache memory comprises one of a group of cache memories comprising: an n-way set associative cache memory, a fully associative cache memory, and a direct mapped cache.

19. A background memory refresher according to claim 17, wherein said block updater comprises:

a way checker, for determining for a given way if data cached in said way is equivalent to the corresponding main memory data; and

a data storer associated with said way checker, for carrying out said updating when said data is not equivalent.

20. A background memory refresher according to claim 19, wherein at least one way of said cache memory comprises a dirty bit to indicate said equivalence status and said way checker is operable to examine said dirty bit of said way.

21. A background memory refresher according to claim 19, wherein said data storer is operable to store said updated data in said main memory at a main memory address associated with said updated data.

22. A background memory refresher according to claim 20, wherein said data storer is further operable to reset said dirty bit when updated data from said way is stored in said main memory.

23. A cache memory background block preloader, for preloading main memory data arranged in blocks into a cache memory, comprising:

a command inputter, for receiving a block initialize command ; and

a cache initializer, for performing blockwise background caching of data of a specified block of main memory into said cache memory.

24. A cache memory background block preloader according to claim 23, wherein said cache initializer comprises:

a cache checker, for determining if data from a selected main memory address is present in said cache memory; and

a data cacher, for carrying out said data caching, if said data is not found to be present.

25. A cache memory background block preloader according to claim 23, said cache memory preprocessor being integrally constructed with a cache memory.

26. A processing system, for processing data from a segmented memory, comprising:

a segmented memory comprising a plurality of memory segments, said memory segments comprising a respective data section and a respective cache memory section;

a processor, for processing data, performing read and write operations to said segmented memory, and for controlling processing system components, and being arranged to access a memory segment via data caching in the respective cache memory section;

a cache memory preprocessor, associated with said processor, for preparing said cache memory sections for use by said processor by performing background processing upon multiple ways of at least one of said cache memory sections in accordance with a multiple-way cache memory processing command received from said processor; and

a switching grid-based interconnector associated with said segmented memory, for providing in parallel switchable connections between said processor and said cache memory preprocessor to selectable ones of said memory segments.

27. A processing system according to claim 26, wherein at least one of said cache memory sections comprises an n-way set associative memory.

28. A method for preparing a cache memory, by:

receiving a cache memory processing command to specify background processing of multiple ways of said cache memory; and

performing background processing upon multiple ways of said cache memory so as to implement said multiple-way cache memory processing command.

29. A method for preparing a cache memory according to claim 28, further comprising controlling communications to said cache memory device according to a priority scheme.

30. A method for preparing a cache memory according to claim 28, wherein said cache memory is arranged in blocks and said command comprises a block update command, and wherein implementing said block update command comprises updating main memory data in accordance with data cached in a specified block of said cache memory.

31. A method for preparing a cache memory according to claim 30, wherein each cache memory block comprises a block of ways, and wherein said updating comprises performing the following steps for each way in said specified block:

determining if data cached in said way is equivalent to corresponding main memory data; and

if said cached data and said corresponding main memory data are not equivalent, storing said updated data in said main memory at a main memory address associated with said cached data.

32. A method for preparing a cache memory according to claim 31, wherein at least one way of said cache memory comprises a dirty bit to indicate said equivalence status and said comprises examining said dirty bit of said way.

33. A method for preparing a cache memory according to claim 32, further comprising resetting said dirty bit when updated data from said way is stored in said main memory.

34. A method for preparing a cache memory according to claim 28, wherein said cache memory is arranged in blocks and said command comprises a block invalidate command, and wherein implementing said block invalidate command comprises invalidating data in a specified block of said cache memory.

35. A method for preparing a cache memory according to claim 34, each cache memory block comprises a block of ways and at least one way comprises a validity bit to indicate a validity status of said way, and wherein said invalidating comprises setting said validity bit of each way in said specified block to invalid.

36. A method for preparing a cache memory according to claim 28, wherein said main memory is arranged in blocks and said command comprises a block initialize command, and wherein implementing said block initialize command comprises caching main memory data of a specified main memory block in said cache memory.

37. A method for preparing a cache memory according to claim 36, wherein each main memory block comprises a block of addresses, and wherein said caching comprises

performing the following steps for each main memory address of said specified block of main memory:

determining if the data of said main memory address is already cached in said cache memory; and

if said data is not cached in said cache memory, caching said data in said cache memory.

38. A method for updating main memory data from cached data, wherein said cache memory is arranged in blocks, by:

receiving a block update cache memory processing command; and

performing background update operations blockwise from a cache memory block specified in said command, so as to update said main memory in accordance with data cached in said specified block within said cache memory

39. A method for updating main memory data from cached data according to claim 38, wherein each cache memory block comprises a block of ways, and wherein said updating comprises performing the following steps for each way in said specified block of ways:

determining if data cached in said way is equivalent to corresponding main memory data; and

if said cached data and said corresponding main memory data are not equivalent, storing said updated data in said main memory.

40. A method for caching main memory data of a main memory into a cache memory, wherein said main memory is arranged in blocks, by:

receiving a block initialize cache memory processing command; and

performing background blockwise caching of data of a main memory block specified in said command into said cache memory.

41. A method for caching main memory data of a specified block of a main memory in a cache memory according to claim 40, each main memory block comprises a block of addresses, and wherein said caching said data into said cache memory comprises performing the following steps for each main memory address of said specified main memory block:

determining if the data of said main memory address is already cached in said cache memory; and

if said data is not cached in said cache memory, caching said data into said cache memory.

42. A program instruction for cache memory block preprocessing, comprising operands defining a cache memory blockwise processing operation and a memory block for performing said processing operation upon, said instruction having low priority so as to prevent the execution of said instruction from interfering with higher priority commands.

43. A program instruction for cache memory block preprocessing according to claim 42, wherein said processing operation comprises a block update operation and said memory block comprises a specified block of ways of a cache memory, and wherein execution of said program instruction comprises updating the data of a main memory in accordance with data cached in said specified block of ways.

44. A program instruction for cache memory block preprocessing according to claim 42, wherein said processing operation comprises a block invalidate operation and said memory block comprises a specified block of ways of a cache memory, and wherein execution of said program instruction comprises invalidating the data in said specified block of ways.

45. A program instruction for cache memory block preprocessing according to claim 42, wherein said processing operation comprises a block initialize operation and said memory block comprises a specified block of addresses of a main memory, and wherein execution of said program instruction comprises caching the data of said specified block of main memory addresses into a cache memory.

46. A compiler configured to support a program instruction for cache memory block preprocessing, for compiling into executable form instruction sequences, said sequences comprising instructions from a predefined set of instructions, wherein said instruction set comprises a cache memory block preprocessing instruction having operands defining a cache memory blockwise processing operation and a memory block for performing said processing

operation upon, and having low priority so as to prevent the execution of said preprocessing instruction from interfering with higher priority commands.

47. A compiler configured to support a program instruction for cache memory block preprocessing according to claim 46, wherein said processing operation comprises a block update operation and said memory block comprises a specified block of ways of a cache memory, and wherein execution of said program instruction comprises updating the data of a main memory in accordance with data cached in said specified block of ways.

48. A compiler configured to support a program instruction for cache memory block preprocessing according to claim 46, wherein said processing operation comprises a block invalidate operation and said memory block comprises a specified block of ways of a cache memory, and wherein execution of said program instruction comprises invalidating the data in said specified block of ways.

49. A compiler configured to support a program instruction for cache memory block preprocessing according to claim 46, wherein said processing operation comprises a block initialize operation and said memory block comprises a specified block of addresses of a main memory, and wherein execution of said program instruction comprises caching the data of said specified block of main memory addresses into a cache memory.